Tracing SRAM Separatrix for Dynamic Noise Margin Analysis under Device Mismatch

Garng M. Huang, Wei Dong, Yenpo Ho, Peng Li

Department of ECE
Texas A&M University
{*ghuang, weidong, ypho, pli*}*@neo.tamu.edu*

Motivation

- **SRAM designs become increasingly susceptible to soft errors.**
  - Various noise injection mechanisms
    - Power supply noises, substrate noises, single event upsets (**SEU**)
- **It is important to evaluate the stability of an SRAM cell.**
  - *static noise margins* (**SNM**)
  - *dynamic noise margins* (**DNM**).
- **It is crucial to evaluate the impacts of process variations.**
  - Introduce significant parameter fluctuations.
  - Lead to asymmetry into SRAM.

Motivation

- **DNM provides a more realistic stability analysis.**
  - SNM finds the maximum tolerable amplitude of ***V*** & ***I***.
  - SNM neglects the important issues.
    - Temporal pattern of the injected noise
    - The SRAM nonlinear dynamics.
- **DNM becomes more involved compared with SNM.**
  - One critical component is the determination of stability boundary (**Separatrix**).
  - The separatrix significantly deviates under the process variations.
  - Requires a full consideration of the complex nonlinear dynamics.

Prior Work

- Statistical process variations have been considered under the context of *static noise margin*.

  [K.Agarwal, S.R.Nassif, DAC06]

- A dynamic noise margin model has been proposed based on the assumption *that the separatrix remains the same even under device mismatch*.

  [B.Zhang, et, al, ICCAD06]

Our Work

- **Rigorously apply a nonlinear system theory to compute the separatrix.**
  - An efficient approach with the SPICE- level accuracy.
  - Only 2 transient analyses on a ***modified*** set of circuit equations.

- **The proposed approach has been implemented in a transistor-level SPICE-like simulator.**

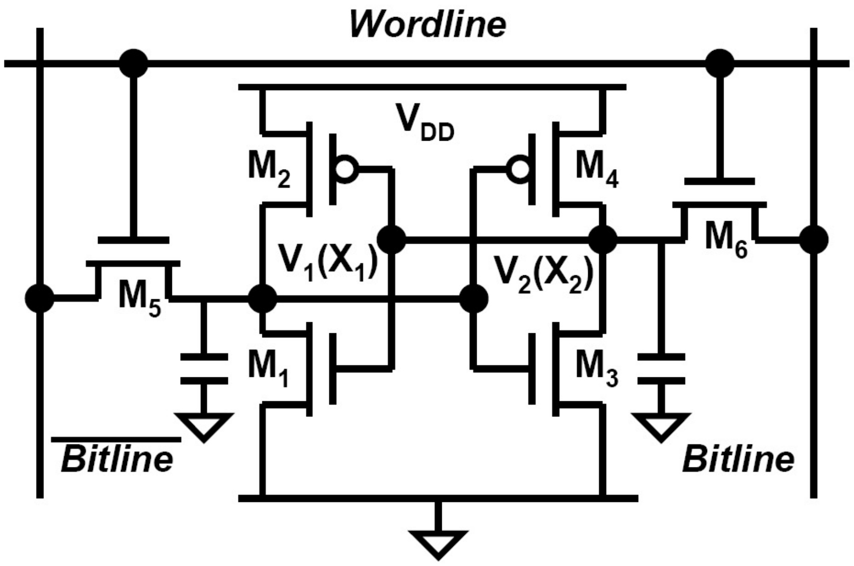
- **Leads to up to 10^3X speedup for the separatrix computation compared to the brute-force method.**

Outline

- Motivation
- Proposed approach
  - Phase portrait of SRAM cell
  - System-theoretical analysis
  - Algorithm to trace the separatrix
- Experimental Results
- Conclusion

Phase Portrait of SRAM Cell

- **The output voltage $V_1(X_1)$ and its complement $V_2(X_2)$ of a standard 6-T SRAM cell form the state space vector of the nonlinear dynamical system.**



A standard 6-T SRAM cell.

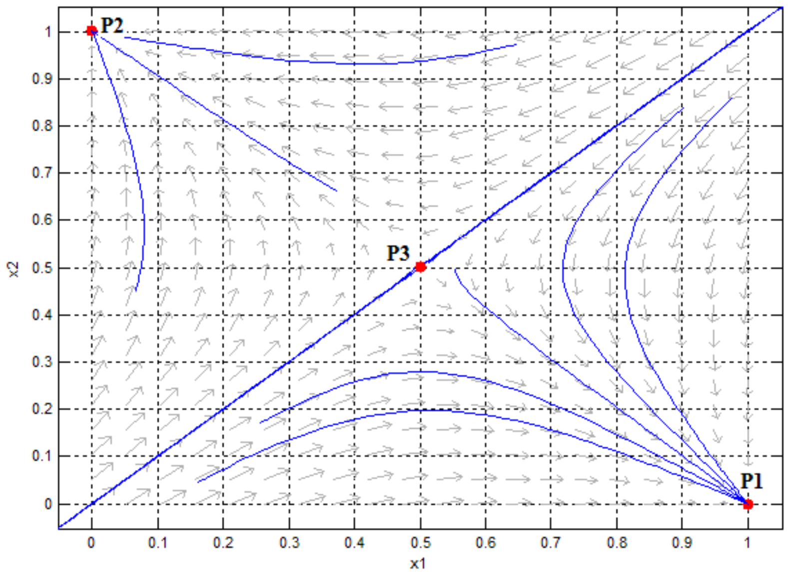
The behavior of the SRAM cell

$$\begin{cases} \partial x_1(t)/\partial t = f(x_1, x_2) \\ \partial x_2(t)/\partial t = g(x_1, x_2) \end{cases}$$

f(*) and g(*) : certain nonlinear functions describing circuit nonlinearities.

Phase Portrait of SRAM Cell

- **A phase portrait on the 2-D x_1-x_2 state space**
  - 2 stable equilibrium (P1 and P2) & 1 meta-stable equilibrium (P3)
  - Separatrix is a 45 degree line passing through the origin and P3.

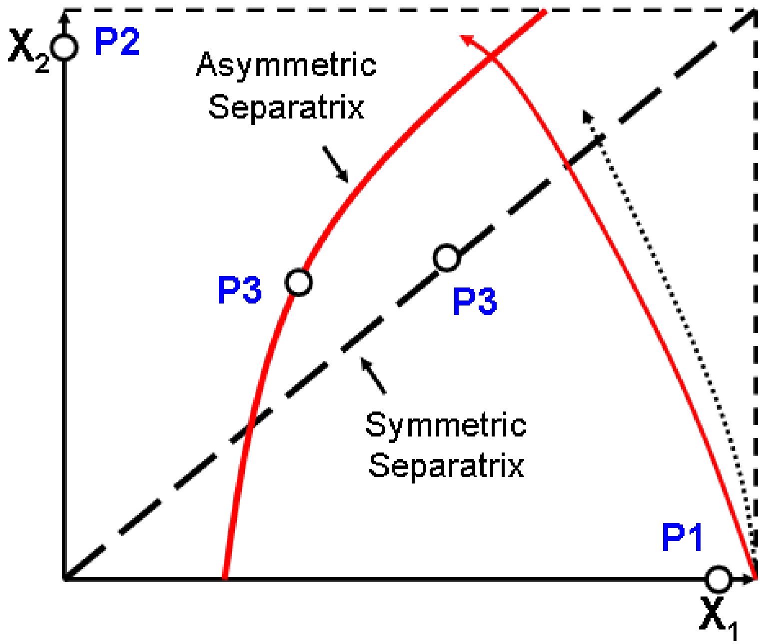


- Supply voltage : 1 V
- P1(x1=1, x2=0), P2(x1=0, x2=1)
- P1 & P2 correspond to the "one" & "zero" states.

- Assume certain noise is injected, the state may be pushed away.
  - *For small noise, the state will be attracted back*
  - *For large noise, the state will flip*

Phase Portrait of SRAM Cell

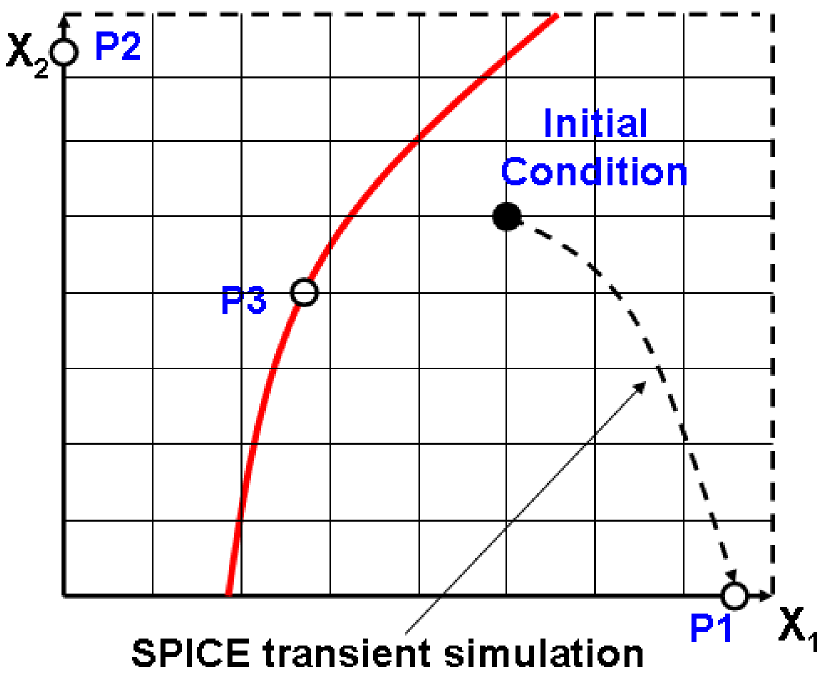
- **Due to unavoidable process variations, the separatrix can deviate significantly from the symmetrical case.**
  - The separatrix does not appear to be a straight line.
    - Errors occur if using linear approximation.
  - Deviation from the linear approximation plays an important role.
    - Any shift of the separatrix will change minimum noise amplitude.



Perturbation of the separatrix and its impact on DNM

Phase Portrait of SRAM Cell

- **A brute-force state space sampling method for finding the separatrix.**
  - Densely sampled using grids.
  - Each grid point is treated as an initial condition.
    - Transient state trajectory ends up at either of the 2 stable equilibriums.



Computationally inefficient due to the large number of transient simulation runs !!

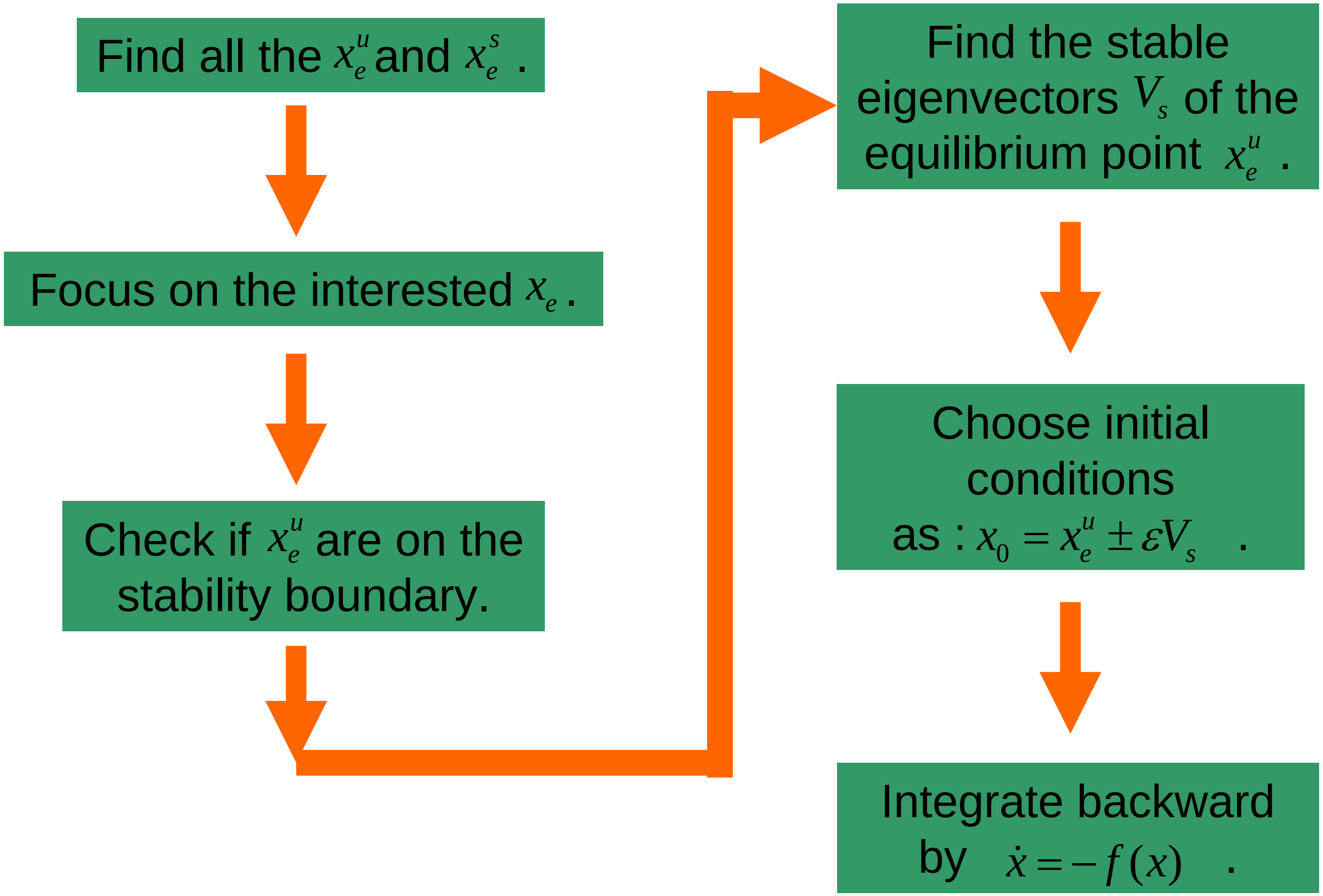
System-theoretical Analysis

- **Equilibrium points :** all the x_e's that satisfy $f(x_e)=0$.
  - A given dynamic equation : $dx/dt=f(x)$

- **Stable Manifold :** $W^s(x_e)=\{x\in R^N \mid \lim_{t\to\infty}\phi(t,x)=x_e\}$
  - $\phi(t,x)$: the trajectory that starts from x and converges to x_e.

- **Stability Boundary (Separatrix) :** $\partial A=\overline{\bigcup W^s(x_e^u)}$
  - x_e^u : the unstable equilibrium on the boundary of A.
  - Start in a small neighborhood of x_e^u along the stable eigenvector directions to integrate reversely to find the stable manifolds.

The Algorithm to Trace the Separatrix



The Algorithm to Trace the Separatrix

Find all the x_e^u and x_e^s .

↓

Focus on the interested x_e .

↓

Check if x_e^u are on the stability boundary.

↓

Find the stable eigenvectors V_s of the equilibrium point x_e^u .

↓

Choose initial conditions as : $x_0 = x_e^u \pm \varepsilon V_s$.

↓

Integrate backward by $\dot{x} = -f(x)$.

Skipped for SRAM case since the unstable equilibrium point is always on the stability boundary.

Skipped if the initial conditions are close to the unstable equilibriums.

Implementation Issues

- To find the unstable equilibrium point of a SRAM design, a nonlinear DC analysis is applied.
- A small perturbation (ΔP) is introduced around the unstable equilibrium.
- Transient analysis is applied to the modified system ($\dot{x} = -f(x)$).
- A small perturbation ($-\Delta P$) in the opposite direction is introduced around the unstable equilibrium.
- A second transient analysis is applied.
- Our algorithm requires one DC analysis followed by two transient analyses.

Experiments

- **The proposed tracing algorithm is implemented as a part of transistor-level SPICE-like circuit simulator.**
  - Using C++ running on Linux platform
  - Level-3 SPICE device model for circuit simulation
  - The device model parameters are listed as below.

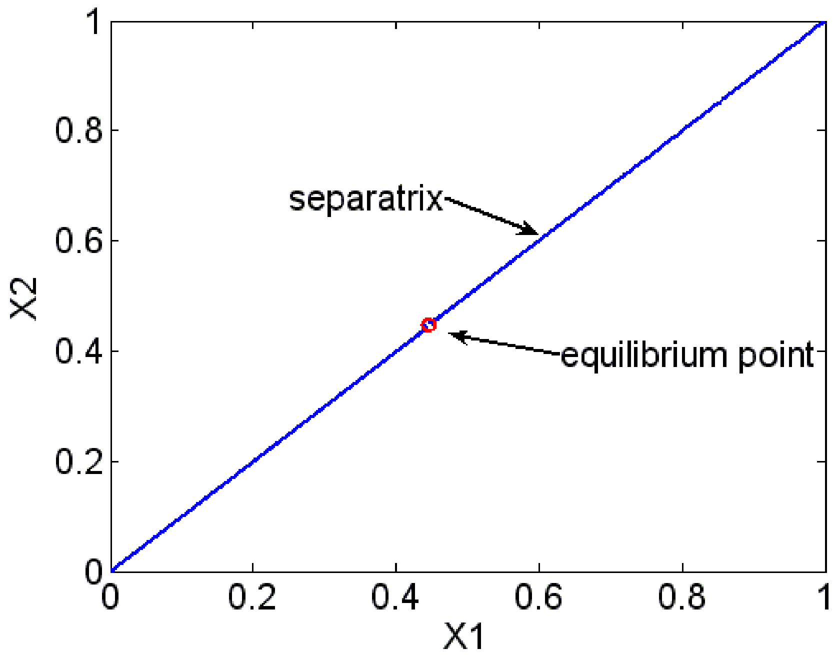
Table-1. 45nm process parameters.

Type	U0	Tox	GAMMA	THETA
NMOS	0.05255	1.75e-9	0.2	0.5
PMOS	0.00696	1.85e-9	0.2	0.3

- **Verify the *accuracy* and *efficiency* of the proposed algorithm for tracing separatrix.**

Experimental Results

- **Find the separatrix of a nominal SRAM design**
  - The meta-stable equilibrium point (0.4465, 0.4465) is determined from the DC analysis.
  - The separatrix is traced using the proposed algorithm.



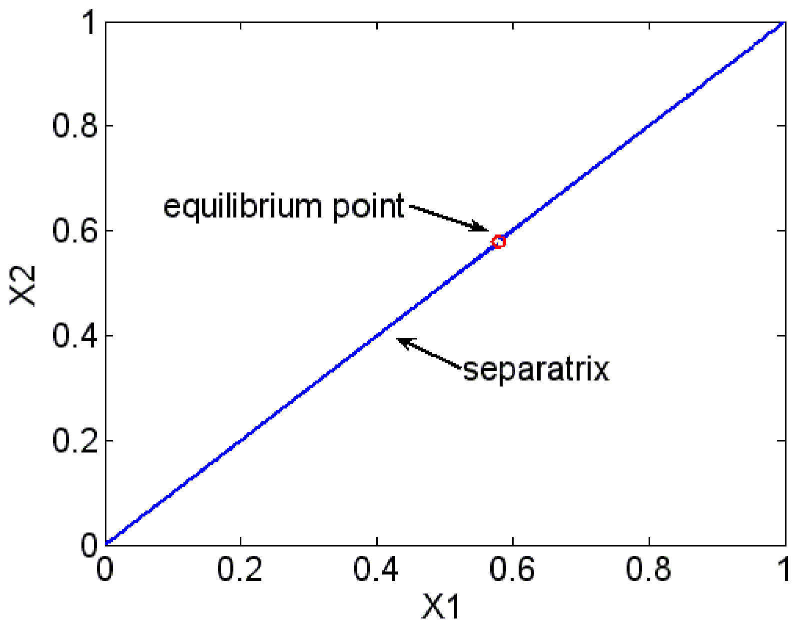
Nominal Case

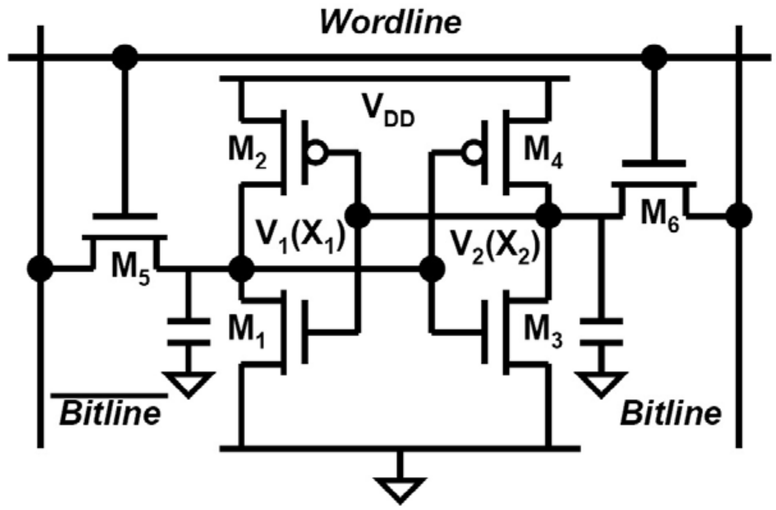
Experimental Results

- **The efficiency and accuracy for our proposed method.**
  - 1 min. for the nominal SRAM case (find the meta-stable equilibrium point and trace the separatrix)
  - The accuracy is mainly determined by the time step control in simulator.
- **The time cost and accuracy for the brute-force method.**
  - Assume one transient simulation costs about 14 secs, 10000 transient simulations need 38 hours (100 x 100 grid).
  - The accuracy is confined by the number of grids used to sample.
- **Our proposed method is much more efficient than the brute-force method.**

Experimental Results

- **Evaluate the impacts of process variations.**
  - Threshold voltage variation
    - Case 1 : The meta-stable equilibrium point (0.5801, 0.5801)





M1: increase by 30%

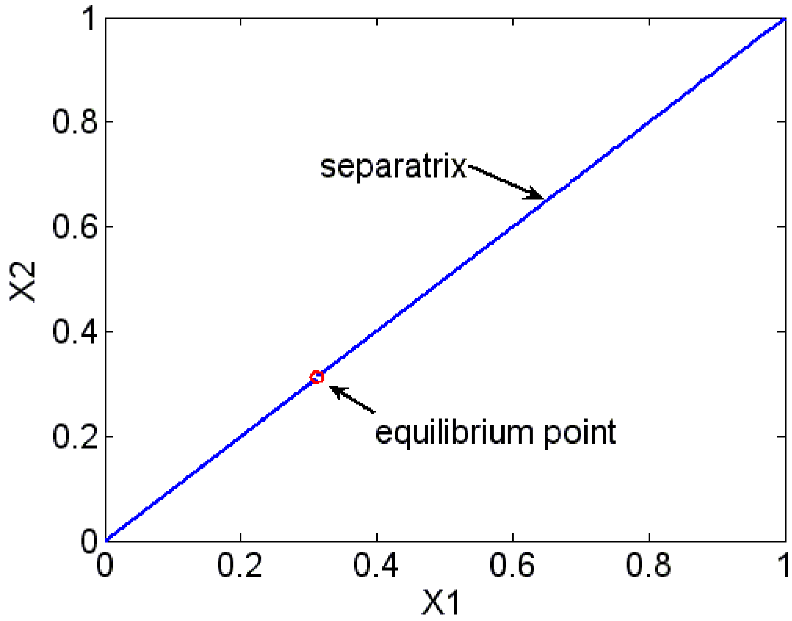
M2: decrease by 30%

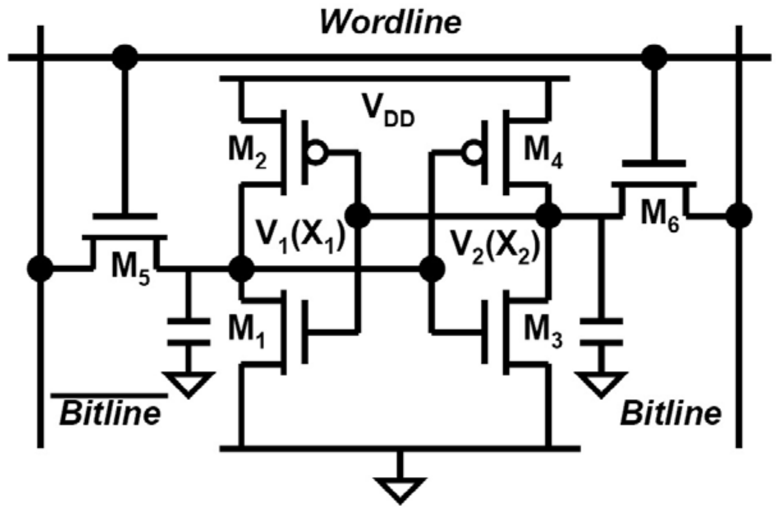
M3: increase by 30%

M4: decrease by 30%

Experimental Results

- **Evaluate the impacts of process variations.**
  - Threshold voltage variation
    - Case 2 : The meta-stable equilibrium point (0.3129, 0.3129)





M1: decrease by 30%

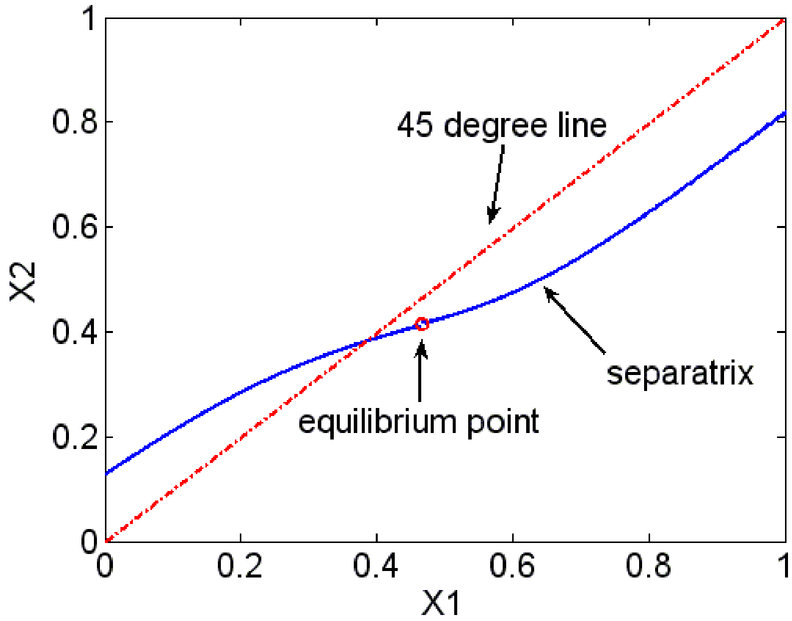
M3: decrease by 30%

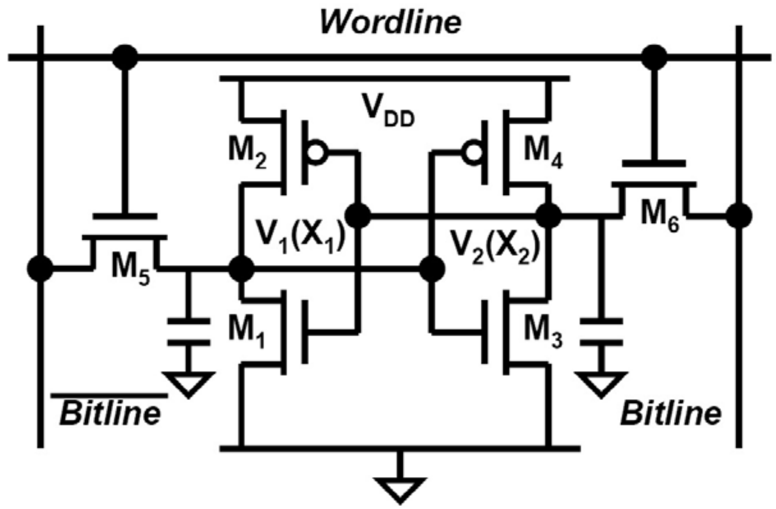
M2: increase by 30%

M4: increase by 30%

Experimental Results

- **Evaluate the impacts of process variations.**
  - Threshold voltage variation
    - Case 3 : The meta-stable equilibrium point (0.4668, 0.4141)



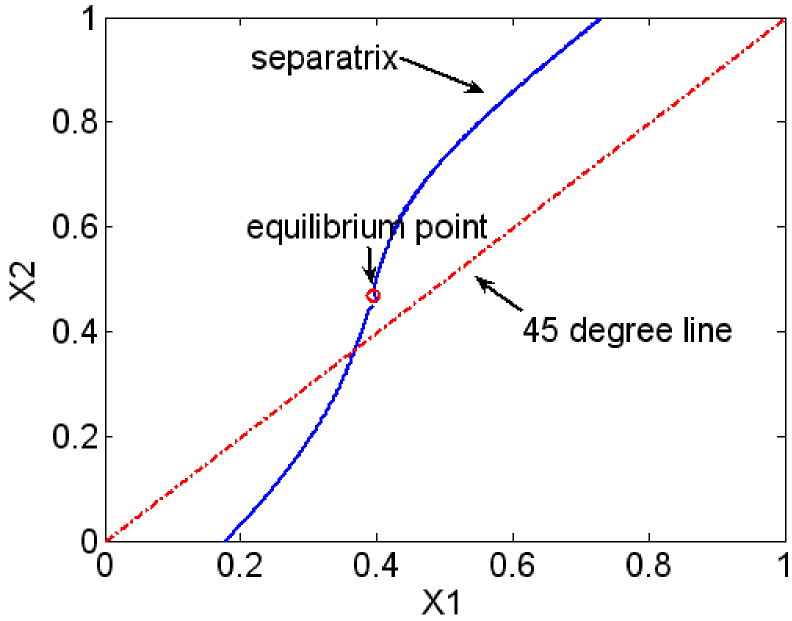


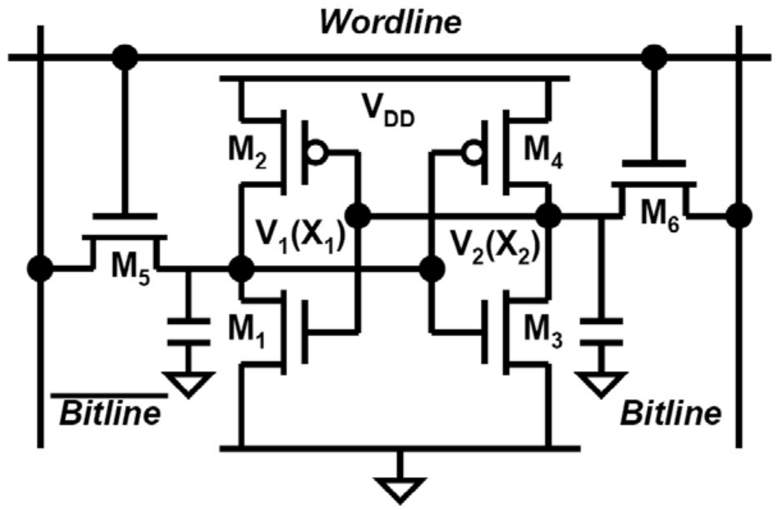
M1: decrease by 30%   M3: increase by 30%

M2: decrease by 30%   M4: increase by 30%

Experimental Results

- **Evaluate the impacts of process variations.**
  - Threshold voltage & effective channel length variation
    - Case 4 : The meta-stable equilibrium point (0.3944, 0.4989)





M1: increase by 30%

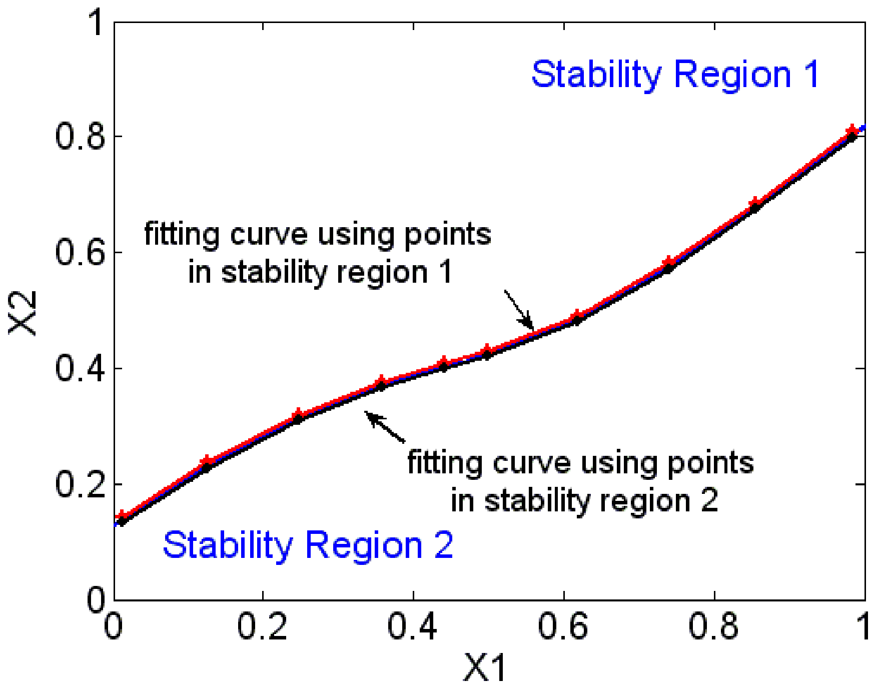
M2: increase by 30%

M3: decrease by 30%

M4: decrease by 30%

Experimental Results

- **Verify the accuracy of the computed separatrix.**
  - Select 10 points from each side of the separatrix.
    - Close to the separatrix.
  - Run transient simulations by taking these points as initial values.
    - Each trajectory ends up at the correct stable equilibrium.



Good accuracy of the computed separatrix !

Conclusion

- **An efficient SRAM stability boundary (*separatrix*) tracing algorithm is presented.**
- **Our proposed algorithm is based on a rigorous nonlinear system theory.**
- **Quickly find the non-ideal separatrix by performing transient analysis based tracing.**
- **Provides useful insight to SRAM dynamic stability margin when process variation is introduced.**

Thanks

Any Question ?